

RECORD COPY

PCT/ I N 2 / 0 0 154

10/500919

Rec'd PCT/PTO 08 JUL 2004

THE PATENTS ACT, 1970

REC'D 04 OCT 2002

WIPO

PCT

It is hereby certified that annexed hereto is a true copy of the  
Provisional and Complete specification filed in respect of Patent  
application No.0020/MAS/2002 dated 08<sup>th</sup> January 2002 by Bluechip  
Infoway Pvt. Ltd., F-101, 1<sup>st</sup> Floor, Suraj Towers, 216/13, III Block,  
Jayanagar, Bangalore - 560011, Karnataka India .....

.....In witness thereof

I have hereunto set my hand

Dated this the 12<sup>th</sup> day of August, 2002  
21<sup>st</sup> day of Sravana, 1924 (Saka)

*K.M. Viswanathan*  
(K.M. VISWANATHAN)

ASSISTANT CONTROLLER OF PATENTS & DESIGNS

PATENT OFFICE BRANCH  
CHENNAI - 600 018

**PRIORITY  
DOCUMENT**  
SUBMITTED OR TRANSMITTED  
BUT NOT IN COMPLIANCE WITH  
RULE 17.1(a) OR (b)

BEST AVAILABLE COPY

FORM 2

THE PATENTS ACT, 1970  
(39 OF 1970)

COMPLETE SPECIFICATION

(See under section 10)

Treated  
as

Provisional  
21.7.01

An Unique versatile executor engine which interprets business language and grows with the needs.

Bluechip Infoway Pvt. Ltd.,  
F-101, I Floor, Suraj Towers,  
216/13, III Block, Jayanagar,  
Bangalore - 560 011  
India.

The following specification particularly describes the nature of the invention and the manner in which it is performed.

WAS 2002

0020

14/1/2002

ORIGINAL

#### Field of Invention:

This invention in general relates to business related software. Further this invention relates to an unique versatile executor engine which interprets business language and grows with the needs.

It is to be noted that the novelty of the invention is focused on executor engine. It is an executor engine with a difference. It can interpret business language offering unlimited options to customize. It is a no failure, never redundant engine that can grow with the needs, saves programming time to great extent (an estimated minimum of 10 fold). Enables that domain expert to build solutions without the help of programmers / Programming team.

The salient feature of the invention lies in the provision to overcome the limitation in the prior art. The steps involved to provide an end to end business solution are various and elaborate, often time consuming and developed at huge costs. At first, the consumer poses his requirements to the domain expert who in turn analyses and structures a solution and passes it on to the programmer / programming team. They then write a program to suit the needs. Such programs will be to a specific process and have limited flexibility. Any further changes if at all could be incorporated has to be made by the programmer / programming team.

The success of the program according to invention depends upon the ability of the programmer / programming team, to comprehend fully both the business process in its entirety and the technical know- how. Such people equipped with both domain expertise and technical expertise is very rare in the industry. So ~~searching such people, hiring team and adding to this the time constraint could~~ only send the manufacturing costs spiraling up, rendering such solution providers less competitive.

#### Prior art:

It is to be noted that none of the existing business process solution providers have an executor interpreter that can interpret business language. So programming skills are a must to maneuver the software. This results in considerable investment in Human resources, Hardware and Software resources.

It is obvious that the existing business process solution providers target specific business process limiting their scope of usage and also their flexibility to accommodate changes. Because of this limited flexibility, such solution providers cannot grow with the needs.

The architecture is not adaptable to change in business trends and very few options are available to customize and often become redundant. The objects of handling core functionalities are huge and hence in a significant measure, renders them less scalable. The code keeps on changing from customer to customer making them more bug prone. The domain expert cannot build solutions without the help of programmers / programming team.

Huge programming time, less flexible, considerable maintenance cost (programmers to be involved at each stage of accommodating a change), failure rate (an estimated 33%), lack of versatility (provides solutions to specific business processes), makes this out of reach to the medium and small scale entrepreneurs.

It has been invented "an unique executor engine" which encompasses and caters to each and every requirement of any business process. It is a versatile, flexible, time saving (estimated to a minimum 10 fold saving in programming time as depicted in the chart below), user friendly, no programming skills required and all importantly no failure and never redundant.

It is the primary object of the invention to invent a novel business solution software having an unique versatile executor engine which interprets business language and grows with the needs.

It is another object of the invention to invent a business solution software which is unique.

It is another object of the invention to invent a business solution software which is versatile and flexible.

It is another object of the invention to invent a business solution software which is time saving and user friendly.

---

It is another object of the invention to invent a business solution software wherein no programming skills are required and no failure is envisaged.

Further objects of the invention will be clear from the following description:

An unique versatile executor engine which interprets business language and grows with the needs comprises an executor engine which is unique and which interprets business language wherein the said software tool has two parts namely BUILDER and EXECUTOR ENGINE, the Builder will capture the business process in terms of inputs and associated processes, outputs in terms of queries and reports according to said rules, the Executor Engine will read the data, interpret the rules and executes the rules such that the architecture is open ended for facilitating easy scalability.

Now the invention will be described in detail in the following complete specification.

The nature of the invention and the manner in which the invention is to be performed is clearly described.

The invention is described in detail with reference to drawings which accompany the complete specification.

Figure 1 of the drawings shows the development cycle using Executor Engine.

Figure 2 of the drawing shows the chart indicating the concept of development of the new software.

Figure 3 shows the time frame for implementation of ERP in the conventional method.

Figure 4 of the drawings shows in detail time frame for customization of ERP.

Figure 5 shows the time frame for the building expertise using PRO FIT - 5RM.

The invention provides a new method to build or to create business solution software.

The software tool has two parts viz., the Builder and the Executor Engine.

The Builder will capture the business process in terms of inputs and associated processes, outputs in terms of queries and reports and users and privileges and rules that are stored in data.

The Executor Engine will read this data, interpret the rules and executes the rules.

The objects handling the core functionalities are small and robust which requires virtually no programming maintenance after a project is completed.

The architecture is open ended thereby facilitating easy scalability. It scales down the steps involved in software development cycle time considerably in addition to providing a no compromise swift solution.

The tool is developed by understanding the nature of different business processes themselves rather than limiting the scope to study specific business processes. The basis is as follows: Any business process will contain a set of related entry documents and processes. The builder is designed in such a way to define entry documents, relationships to other documents and associated processes by domain experts.

The Executor Engine reads the definition, provides the end user with a user interface, accepts data, stores them in the data base as per the definition. So, the Executor Engine is the same at all customer sites. Hence, the solutions are more tested, robust and cost effective.

Coding is completely eliminated making it less bug prone.

The chart accompanying the specification depicts comparative advantage of our "PRO FIT - 5RM" with prior art as it exists today. It clearly depicts the shortening of the software development cycle to a great extent, programming efforts virtually and in particular the time saved.

#### Chart

The software engine invented by us will cut short the software development cycle to a great extent. **[FIG. 2 SHEET NO. 2]**

Programming efforts will be removed completely from the development cycle.

The time saving could be described as follows:

What can be done by an experienced programmer in a months time can be done by a non - programmer in just a days time. The time saving ratio is 1:24.

Following is the prior art in software development cycle: **[FIG. 1 SHEET NO.1]**

(The time estimate is for a mid size ERP project)

#### Advantages:

1. The Engine can interpret business language, thus enabling the domain experts to build solutions without the help of programmers / programming team.
2. Time saving will be significant (10 fold) compared to the traditional solution providers.
3. The executor can build solutions which can grow with needs making it highly flexible, never redundant.
4. Failure is nil - compared to traditional ones with an estimated 33% failure.
5. Solutions to any business process and not catering to specific business process making it a versatile engine for solution providers.
6. Virtually nil program maintenance after project completion, which emphasizes

easy maintenance.

7. It enables to build any business process as envisaged by the domain expert in total without any limitations. In other words, it offers unlimited options to customize it to specific needs.
8. It is designed to fulfil the appetite for all types of information that is difficult to envisage at one go and its need crops up and are needed at crucial times when decisions have to be based on them.
9. The versatility of the invention is obvious.

Application: In all fields of business.

Identification : PRO FIT - 5RM.

Time frame for implementation of ERP (conventional method): **[FIG. 3 SHEET No.3]**

This methodology includes development of the modules for a small manufacturing enterprise right from system study, design and complete coding. The basic modules covered are financial accounting, inventory, production planning and control, order processing and payroll.

Time frame for customization of ERP: **[FIG. 4 SHEET No.4]**

This methodology includes customization of basic existing modules for a small manufacturing enterprise. The enhancements could be achieved only by modifying the program. The basic modules covered are financial accounting, inventory, production planning and control, order processing and payroll.

Time frame for building enterprise solutions using PRO FIT - 5RM: **[FIG. 5 SHEET No.5]**

This methodology includes customization of the complete requirements for a small manufacturing enterprise without programming. The enhancements could be achieved merely by modifying the solutions but not coding. The basic modules covered are financial accounting, inventory, production planning and control, order processing and payroll.

## ABSTRACT

An unique versatile Executor Engine which interprets business language and grows with the needs comprises an Executor Engine which is unique and which interprets business language wherein the said software tool has two parts namely BUILDER and EXECUTOR ENGINE. The builder will Capture the business process in terms of inputs and associated processes, outputs in terms of queries and reports according to said rules, the executor engine will read the data, interpret the rules and executes the rules such that the architecture is open ended for facilitating easy scalability.

For **BLUECHIP INFOWAY PVT. LTD.**



Director

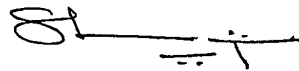
Signature of the applicant



**WE CLAIM :**

1. An unique versatile Executor Engine which interprets business language and grows with the needs. The invention comprises an Executor Engine which is unique and which interprets business language wherein the said software tool has two parts namely the Builder and Executor Engine. The Builder will capture the business process in terms of inputs and associated processes, outputs in terms of queries and reports according to said rules. The Executor Engine will read the data, interpret the rules and executes the rules such that the architecture is open ended for facilitating easy scalability.
2. An unique versatile engine as described in the complete specification and as illustrated by way Of Drawings.

**For BLUECHIP INFOWAY PVT. LTD.**

R. 

**Director**

Dated this 4<sup>th</sup> day of January 2002

Signature of the applicant

**Prior Art: Time Estimate for a Midsize ERP project**

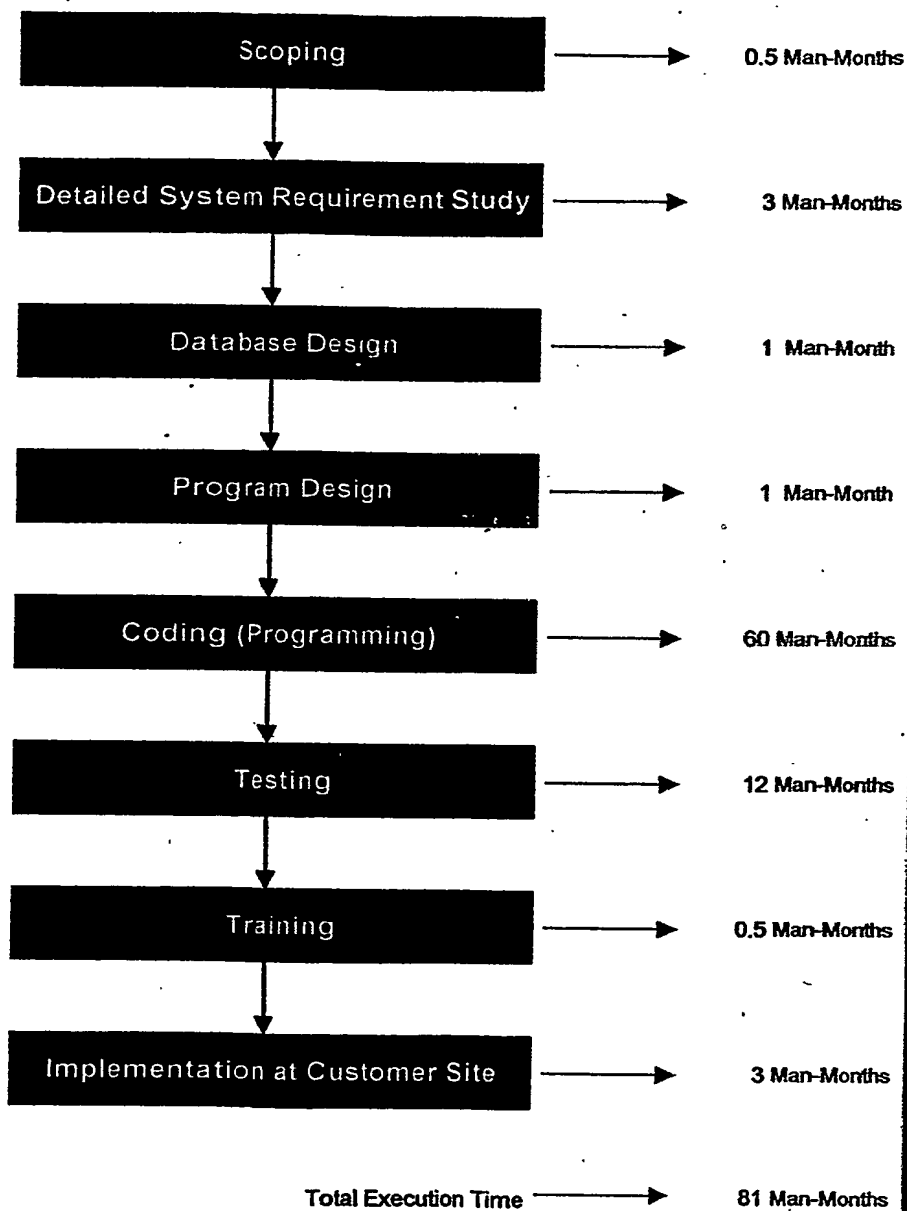
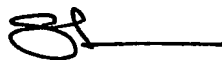



Fig. 1

For Bluechip Infoway Pvt. Ltd.

R.    
(Director)

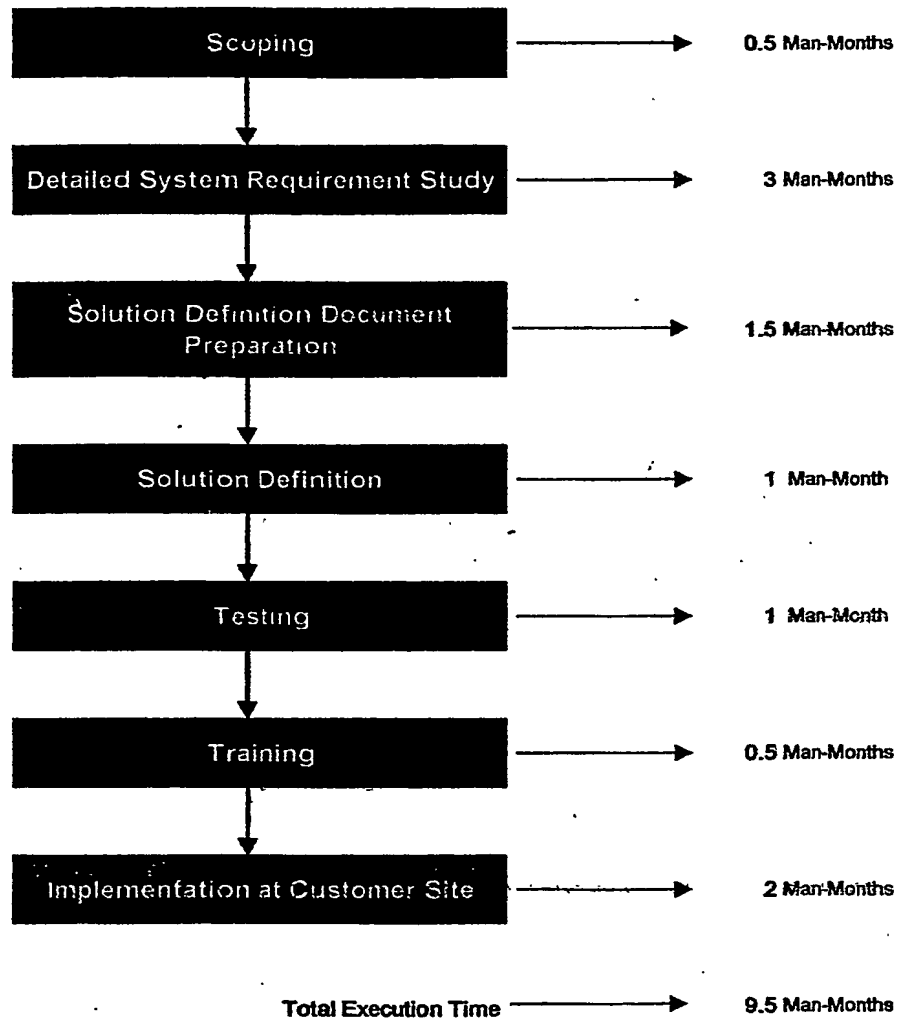
**DEVELOPMENT CYCLE USING THE 'EXECUTOR ENGINE' OF PROFIT 5RM**

Fig. 2

The engine has two layers

**BUILDER****EXECUTOR**

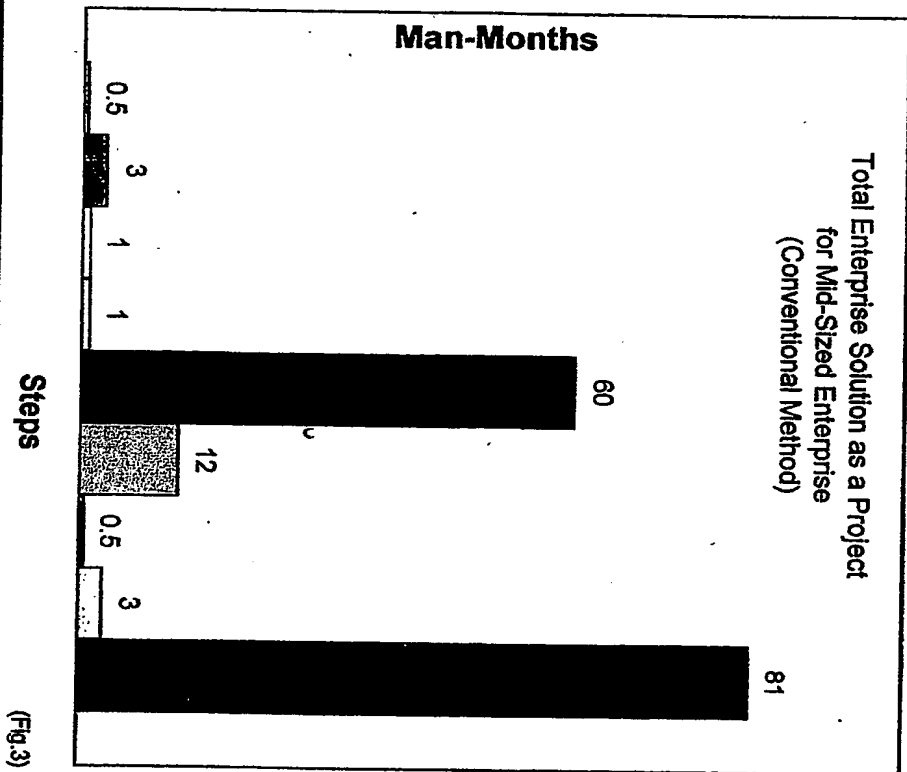
The Builder accepts Definition and stores it in a database.

The EXECUTOR ENGINE understands the Definition and executes it.

For Bluechip Infoway Pvt. Ltd.

R. S. [Signature]

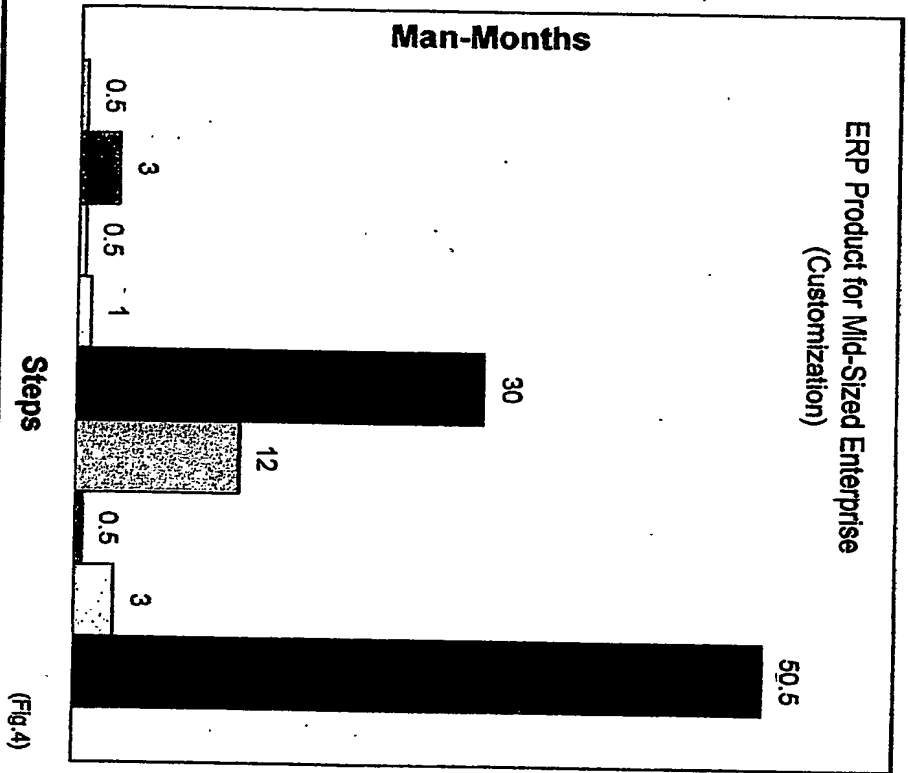
(Director)



- ☒ Scoping
- ☒ Detailed System Requirement Study
- ☐ Database Design
- ☐ Program Design
- ☒ Coding (Programming)
- ☒ Testing
- ☒ Training
- ☐ Implementation at Customer Site
- ☒ Total Execution Time

For Bluechip Infoway Pvt. Ltd.

R. B.                       
(Director)

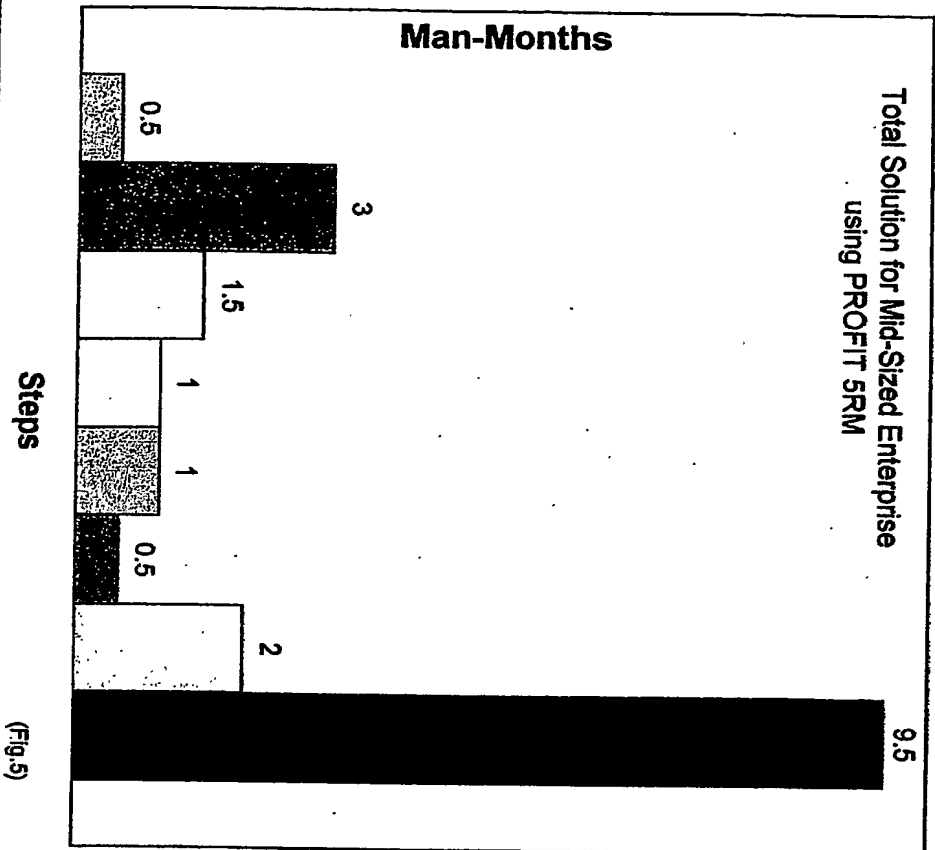


(Fig 4)

- ☒ Scoping
- ☒ Detailed System Requirement Study
- ☐ Gap Analysis (Gap B/w. the requirement & the ERP Software)
- ☐ Process Re-Engineering Meetings
- ☒ Customising the ERP product
- ☐ Testing
- ☐ Training
- ☐ Implementation at Customer Site
- ☒ Total Execution Time

For Bluechip Infoway Pvt. Ltd.

*R. B.*  
(Director)



- Scoping
- Detailed System Requirement Study
- Solution Definition Document Preparation
- Solution Definition
- Testing
- Training
- Implementation at Customer Site
- Total Execution Time

For Bluechip Infoway Pvt. Ltd.

2.                       
(Director)

20/MAS/2002  
08/01/02

Complete After Provisional  
Left 31 JUL 2002

ORIGINAL

## FORM 2

THE PATENTS ACT, 1970  
(39 OF 1970)

### COMPLETE SPECIFICATION

(See under section 10)

**An Unique versatile executor engine which interprets business language and grows with the needs.**

**Bluechip Infoway Pvt. Ltd.,  
F-101, 1<sup>st</sup> Floor, Suraj Towers,  
216/13, III Block, Jayanagar,  
Bangalore - 560 011  
India.**

The following specification particularly describes the nature of the invention and the manner in which it is performed.

## **Field of Invention**

This invention in general relates to the field of building information systems for enterprises. Further, this invention relates to a unique versatile executor engine, which can interpret & execute transaction structures & information views. Hence, the information system thus built can grow along with the needs of the enterprise.

The novelty lies in the new methodology that has been evolved to build information systems with the executor engine.

An information system built in this methodology will allow online and on site creation of new options and changing the existing options without any down time.

Software programs are not written to build an information system in this methodology. The information system is analyzed and split into transaction structures & information views. These are then defined using the PROFIT 5RM builder, which stores these definitions as data. This is interpreted & executed by the executor engine to provide the necessary information system. The architecture of the system is described in fig 1.

Hence, an information system can be built in a very short time (refer to fig 4,5 & 6). An estimated minimum of 1:50 time saving can be achieved in this methodology in comparison with the prior art.

Enables domain experts to build information systems without the help of programmers or programming team.

The salient feature of the invention lies in the provision to overcome the following limitations in the prior art.

The steps involved to provide an end to end business solution are various and elaborate, often time consuming and developed at huge costs (Refer to fig 2).

At first the consumer poses his requirements to the domain expert who in turn analyses and structures the solution and passes it on to the programming team. They then write a program to suit the needs. Such programs will be to a specific process and have limited flexibility. Any further changes, if at all to be incorporated, has to be made by the programming team.

The success of the program according to the invention depends upon the ability of the programming team, to comprehend fully both the business process in its entirety and the technical know how. Such people equipped with both domain expertise and technical expertise is very rare in the industry. So, sourcing such



people, hiring the team and the time constraint could only send the developing costs, spiraling up, rendering such solution providers less competitive.

### Prior art

It is to be noted that none of the existing business process solution providers have an executor interpreter that can interpret transaction structures and information views. So, programming skills are a must to maneuver the software. This results in considerable investment of human resources, hardware and software resources.

It is obvious that the existing business process solution providers target specific business processes limiting their scope of usage and also their flexibility to accommodate changes. Because of this limited flexibility, such solution providers cannot grow with the needs on time.

The architecture is not adaptable to change in business trends and very few options are available to customize and often become redundant. The objects of handling core functionality are huge and hence in a significant measure render them less scalable.

The code keeps on changing from customer to customer making them more bug-prone. The domain expert cannot build solutions without the help of programming team.

Huge programming time, less flexible, considerable maintenance costs (programmers to be involved at each stage of accommodating change), high failure rate, lack of versatility (provide solutions to specific business needs) makes it very difficult to build and maintain information systems.

It has been invented a "unique executor engine" which encompasses and caters to each and every requirement of any business process. It is versatile, flexible, substantial time and cost-saving, user friendly, easily scalable, no programming skills required and all importantly no failure and never redundant.

It is the primary object of the invention to invent a unique versatile executor engine, which interprets & executes the transactions structures and information views.

It is another object of the invention to invent a new methodology, which simplifies business software development.

It is another object of the invention to invent a unique executor engine, which is versatile and flexible.

It is another object of the invention to invent a unique versatile executor engine which is time saving (substantial).

It is another object of the invention to invent a unique versatile executor engine, which is cost saving (substantial).

It is another object of the invention to invent a unique versatile executor engine, which is user friendly.

It is another object of the invention to invent a unique versatile executor engine wherein no programming skills are required.

Further objects of the invention will be clear from the following description:

A unique versatile executor engine which interprets transaction structures and information views where in the said software tool, has two parts namely builder and executor engine. The builder will capture the business process in terms of transaction structures with process maps and information views. The executor engine will read and execute the definitions. The architecture of the executor engine has been made in such a way that scaling to new technologies can be made with ease without affecting the existing information system.

Now the invention will be described in detail in the following complete specification.

The nature of the invention and the manner in which the invention is to be performed is clearly described through drawings, which accompany the complete specification.

## **PROFIT 5RM ARCHITECTURE**

Refer to Figure 1 for the open ended architecture diagram of PROFIT 5RM.

A domain expert interacts with the end user & understands the information system required by the organization.

The information system is analyzed and made into simple structures called **Transaction Structures** and **Information Views**. These are defined using the PROFIT 5RM Builder.

The Builder stores the definitions of Transaction Structures & Information Views as data into the database.

The Graphical User Interface Layer (GUIL) presents a user interface for the end user based on the Transaction Structure.

The GUIL interacts with the end user and submits various requests for transaction processing or information request to the **Process Request Server [PRS]**.

Process Request Server (PRS) provides services to process transactions and information requests. The **Transaction Structures** govern the transaction processing done by the PRS. The **Information Views** govern the processing of information requests.

It is important to note that the PRS does not have any programs particular to any business process.

In this approach, software programs are not written to make an information system. Instead, an information system is made into Transaction Structures and Information Views and stored in the database.

### **Transaction Structures [TStruct]**

Transaction Structures are the building blocks of an information system. These govern the data input, validation and storage functionality of an information system.

A transaction structure is a collection of

- Data containers
- GUI Definitions
- Validation statement
- Process maps
- Print formats

The executor engine interprets a Transaction Structure and a transaction form is presented to the end user. The end user can input data into the information system through these forms.

**Data Containers** are logical groups of input data in a transaction structure. Containers can be classified into simple and tabular. A simple container will hold input fields which will hold only one value in a single transaction. A tabular container will hold input fields in the form of a grid and hence can hold many values in a single transaction. A transaction structure can have any number of containers.

**Input fields** are elements, which hold a piece of data of a transaction. Each container can have one or more input fields.

An input field is identified through a name that is unique in a transaction structure.

Input fields can be categorized into simple, calculated, auto-generated and referral fields.

Simple input fields accept data into it based on the type definition.

Calculated input fields, evaluate an expression to get its value. The expression could be based on any field in the TStruct.

Auto-generated fields generate serial numbers for each transaction.

Referral fields allow user to pick data from other transactions. Defining SQL statements, as part of the input field does this. The result set of the SQL will be presented to the end user for selection.

**GUI Definitions** can be made to set the look and feel of the transaction forms.

The **Validation Statement** is an expression, which controls the validity of each transaction entered through the transaction form.

**Process maps** provide a way of relating transaction structures and passing data from one transaction to another.

Three kinds of process maps are required for an information system:

- Transaction Generator
- Master-Detail
- Transaction-Group

**Transaction Generator** maps a source transaction structure to a target transaction structure. Such a mapping will create one or more target

transaction automatically when the source transaction is stored. Mapping fields in the source to that in the target does it.

**Master-Detail** maps are made to update data from a detail transaction into another related transaction called the master transaction. The field in the details is mapped to a field in the master along with the type of update to be done.

**Transaction-Group** maps represent a many-to-many user-controlled relationship between transactions from many transaction structures. Transactions from the different structures that need to be related are posted to a common transaction structure. This common transaction structure is called the **Group Structure**.

The transactions in a Group structure can be grouped based on the relationship to be established.

Each group is a collection of transactions. The first transaction in the group is called the **group header** and the subsequent transactions are called the **group elements**. An input field in the group structure is identified as the **Group Validation Field (GVF)**. A group is valid only if the value in the GVF in the group header is greater than or equal to the sum of the GVF values in the group elements.

**Print formats** are text documents with input fields embedded in it. A transaction form can be printed along with the data entered in it. The format of the printed document will be the same as the print format with the input fields replaced with actual data.

### **INFORMATION VIEWS [IView]**

An Information View is a set of related SQL statements that provide data to a view format. It governs the information presentation function of an information system.

Information views are of two types

- Tabular
- Free form

The classification is based on the format in which the view is to be presented.

A **Tabular View** consists of a set of SQL statements. The SQL result sets are related by defining a relation field. Rows in the result sets that have the same value in the relation field are collated into one row & presented in the IView.

Every column in each result set is represented as a column in the IView. The IView can have additional columns which may be computed based on an expression. The expression may include conditional statements also.

Columns in the IView can be hidden to show only relevant information.

The rows in the IView can be grouped based on a GROUP FIELD. All rows that have the same value in the group field are collated & presented as one row in the IView.

Multi level control breaks can be defined in an IView by defining key columns. A total line is printed when the value in the key column changes. Further, report balancing can be defined on any control break. In such a case, a closing and opening balance is automatically printed.

The features discussed above, facilitate making of all kinds of tabular reports that may be required in an information system without writing programs to achieve the same.

A Free Form View consists of a set of SQL statements and a text document with print fields embedded in the document.

Print fields are replaced with actual data before printing the report. The actual data are drawn from the SQL result sets.

The executor engine reads an IView, executes the SQL statements and prepares the IView and presents it as an Information form to the user.

The Information form provides features to navigate through the IView and select any row for further drill down.

A drill down facility can be achieved by linking two IViews. Any column from the selected row can be set to provide parameter values to the SQL statements in the linked IView.

A row in the IView may also be linked to a transaction form. In such a case, the transaction form will be displayed by the executor engine with data related to the selected row in the IView.

## Charts

Figure 1 Sheet 1 of the drawings shows the open-ended architecture of **PROFIT 5RM**.

Figure 2 Sheet 2 of the drawings shows the development cycle as per the prior art, which includes requirement study, system design, database design, coding, testing, training and implementation.

The software executor engine invented by us will cut short the software development cycle to a great extent. Programming efforts will be removed completely from the development cycle. The time saving could be described as follows: What can be done by an experienced programmer in a month, can be done by a non-programmer in just a few hours. The time saving ratio is 1:50. Figure 3 Sheet 3 of the drawings shows the development cycle with **PROFIT 5RM**.

Figure 4 sheet 4 shows the time frame for building an information system in conventional method. The methodology includes development of the modules for a small manufacturing enterprise right from system study, design & complete coding. The basic modules covered are financial accounting, inventory, production planning and control, order processing and payroll.

Figure 5 Sheet 5 shows in detail the time frame for building an information system with products developed in the prior art. This methodology includes customization of basic existing modules for a small manufacturing enterprise. The enhancements could be achieved only by modifying the program. The basic modules covered are financial accounting, inventory, production planning and control, order processing and payroll.

Figure 6 Sheet 6 shows the time frame for building an information system using **PROFIT 5RM**. This methodology includes definition of the complete requirements for a small manufacturing enterprise without programming. The enhancements could be achieved by merely modifying the definitions but not coding. The basic modules covered are financial accounting, inventory, production planning and control, order processing and payroll.

The invention provides a new method to build and create an information system.

## **Advantages of PROFIT 5RM architecture**

### **Domain Neutral**

Information systems for any kind of enterprise, industry or organization can be built using Profit5RM.

### **Time**

This methodology completely eliminates software programming. Hence, an information system can be built in a very short time and with ease.

### **Cost**

A domain expert can build an information system without using a programming team and hence, the built information system will cost much less.

### **Maintenance**

The transaction structures & information views that are the building blocks of the information system are stored as data and hence can be maintained with ease.

### **Change Management**

Changes to processes can be done on the fly and at site because changes to processes now mean changes to transaction structures and information views that are stored as data.

### **No Programming Required**

Domain experts need not depend on a programming team get an information system in place with this methodology as there is no programming involved. Hence a solution built on this platform will have an extremely high rate of success.

### **Easy Scalability**

The PRS services are simple objects, which can be extended to support new technologies by wrapping them in appropriate interface classes. For example these services may be made into Web services by creating SOAP (Simple Object Access Protocol) interfaces. Hence, this layer can be scaled to any new technology with ease and speed.

As business processes are stored as data, scaling the PRS will not affect the information system that has been already defined.

Similarly, altering the definitions immaterial of the technology extensions made to the PRS can change the business processes themselves.



The architecture facilitates PROFIT 5RM to act as an information system foundation for any enterprise.

**Application:** In building information systems for all lines of businesses.

**Identification:** PROFIT 5RM.

## CLAIMS

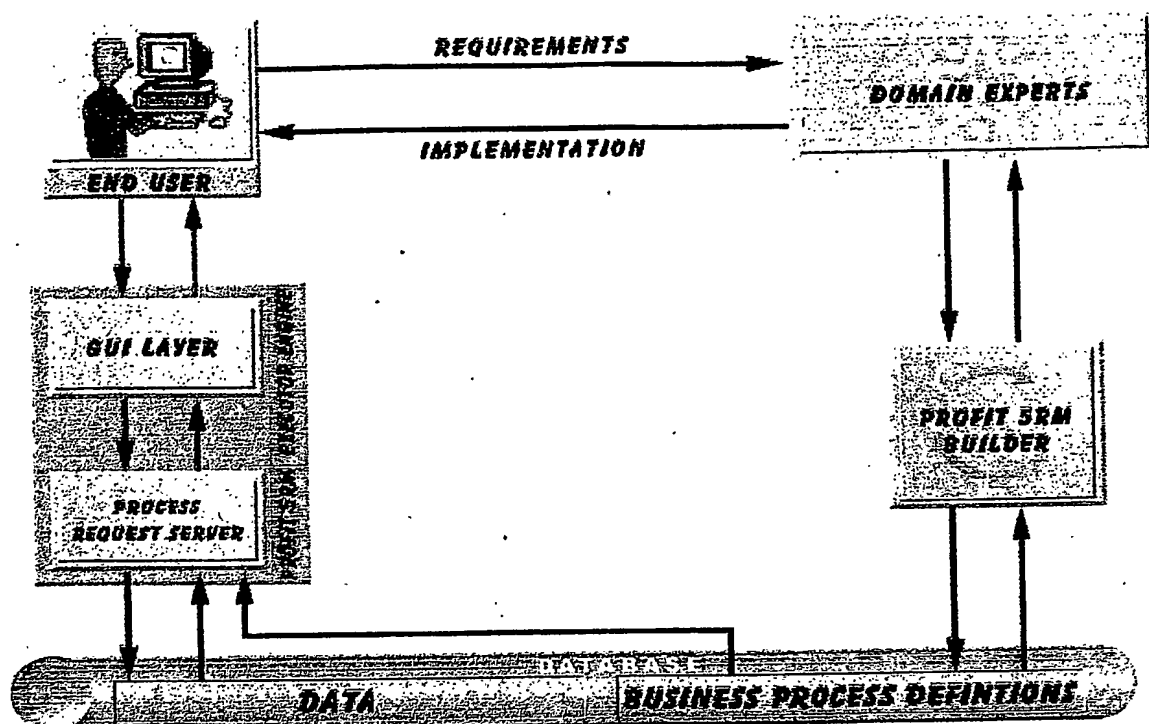
1. We claim for the unique technology, which enables an information system to be built by representing the system requirements as Transaction structures and Information views and storing them as data, instead of writing software programs to achieve the same. Thus making information system building simple and quick.
2. We claim for the open-ended architecture of PROFIT 5RM (as illustrated in Fig 1). This facilitates easy scalability to new technologies without affecting the already made information structure. Further, allows change in the information structure immaterial of the technology extensions made.
3. We claim for the unique versatile executor engine, which interprets & executes the transaction structures and information views.
4. We claim for the technology, which reduces software development cycle time in a ratio of 1:50 compared to the traditional methodology.
5. We claim for the technology, which is domain-neutral and hence that can be used to build information systems for all domains.
6. We claim for the technology, which facilitates onsite as well as online development and which can be maintained with ease.

## **ABSTRACT**

Profit5RM incorporates a unique technology, which enables an information system to be built without writing software programs. This makes information system building simple and quick.

The information system is analyzed and split into transaction structures & information views. These are then defined using the PROFIT 5RM builder, which stores these definitions as data. This is interpreted & executed by the unique executor engine to provide the necessary information system. The architecture of the system is described in fig 1.

# PROFIT 5RM ARCHITECTURE



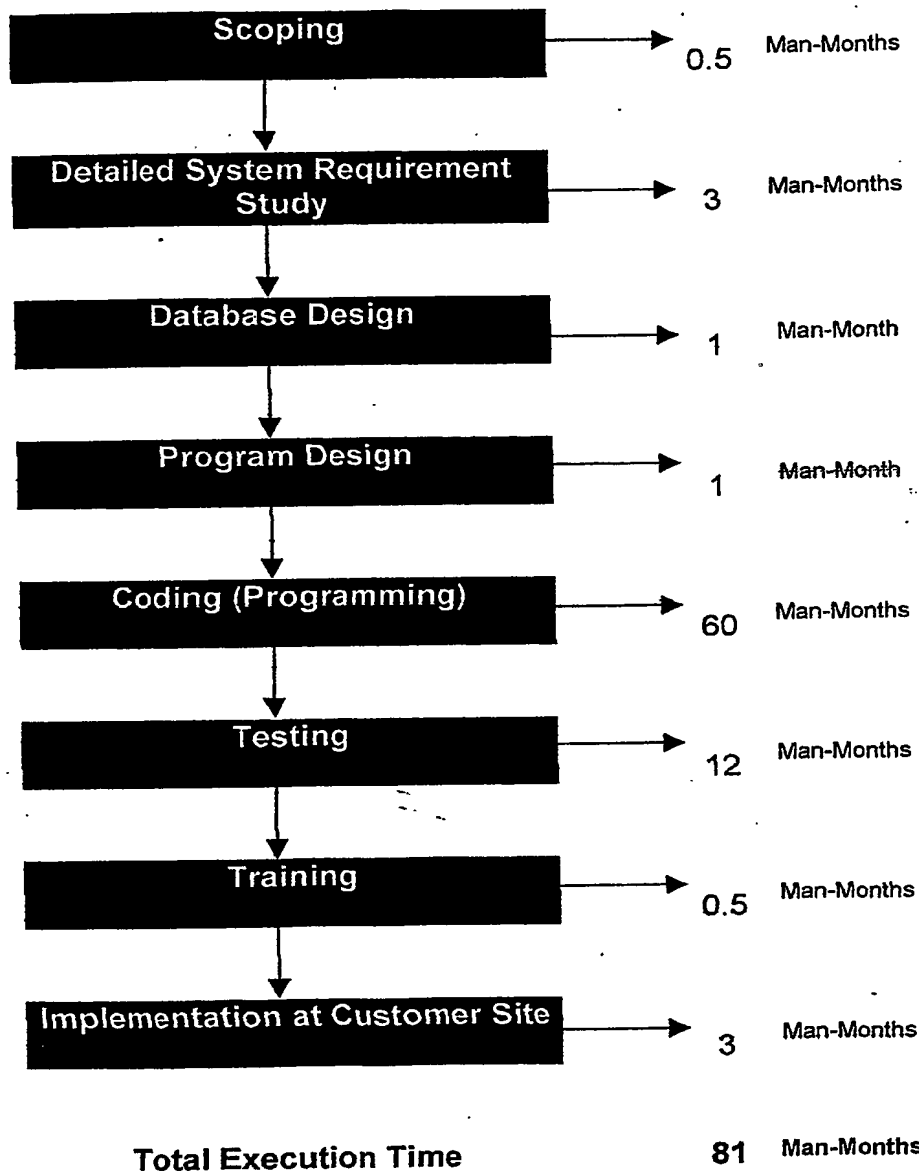
For BLUECHIP INFOWAY PPT. 1020

R. 88

100

2/6

**Prior Art: Time Estimate for building an Information System for a  
Midsize organization**

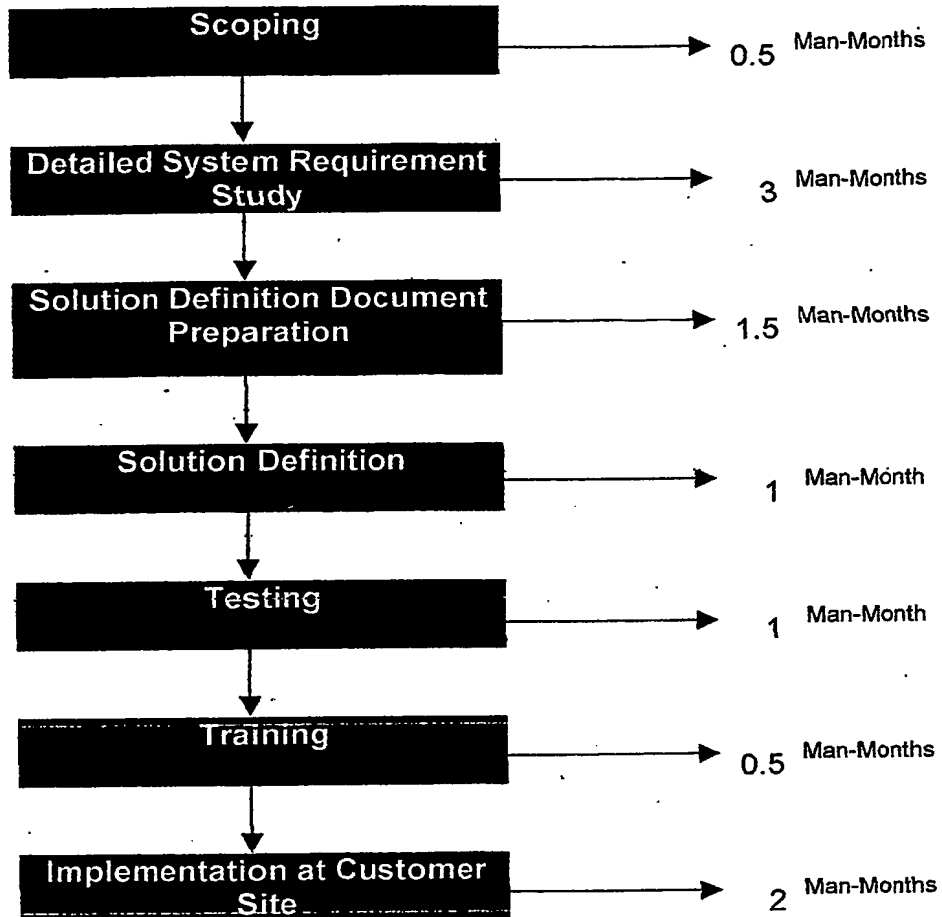


For BLUECHIP INFOWAY PVT. LTD.

R. [Signature]  
Director

3/6

**PROFIT 5RM: Time Estimate for building an Information System for a Midsize organization**



**Total Execution Time 9.5 Man-Months**

**For BLUECHIP INFOWAY PVT. LTD.**

**R. S. T**

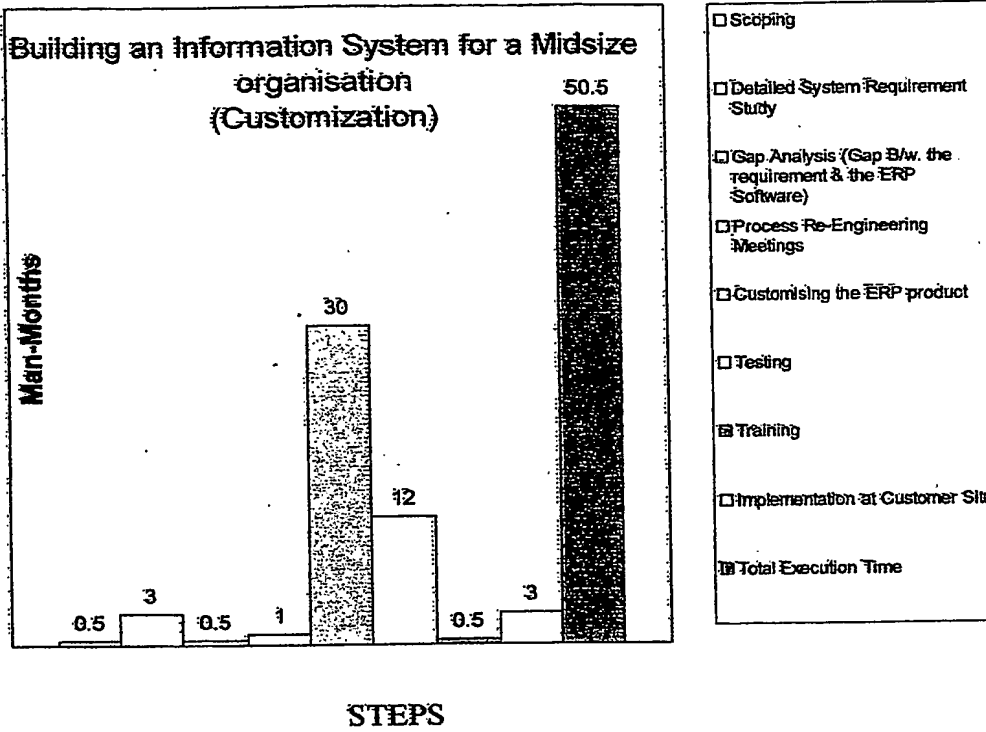
**Director**

Phase	Man-Months
Scoping	0.5
Detailed System Requirement Study	3
Solution Definition Document Preparation	1.5
Solution Definition	1
Testing	1
Training	0.5
Implementation at Customer Site	2
<b>Total Execution Time</b>	<b>9.5</b>

FOR BLANCHIFF INFORMATION PTE. LTD.

R. OL IT

5/6

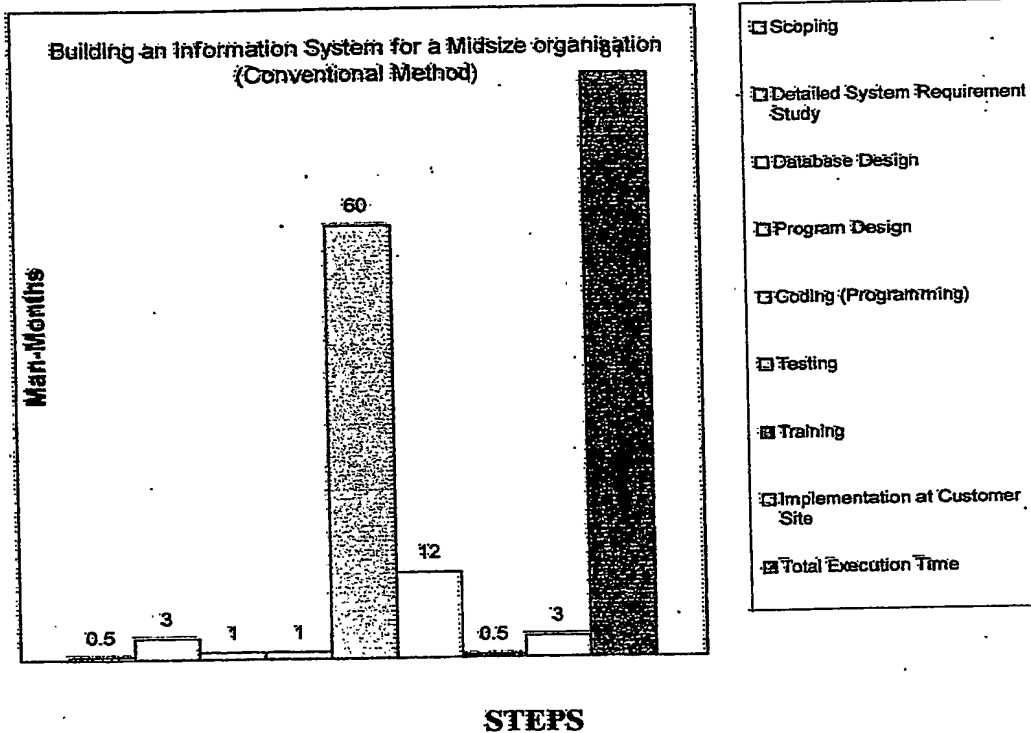


For BLUECHIP INFOWAY PVT. LTD.

R. [Signature]  
Director



6/6



For BLUECHIP INFOWAY PVT. LTD.

R. S. T.  
Director

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**